

Nathaniel Serrano

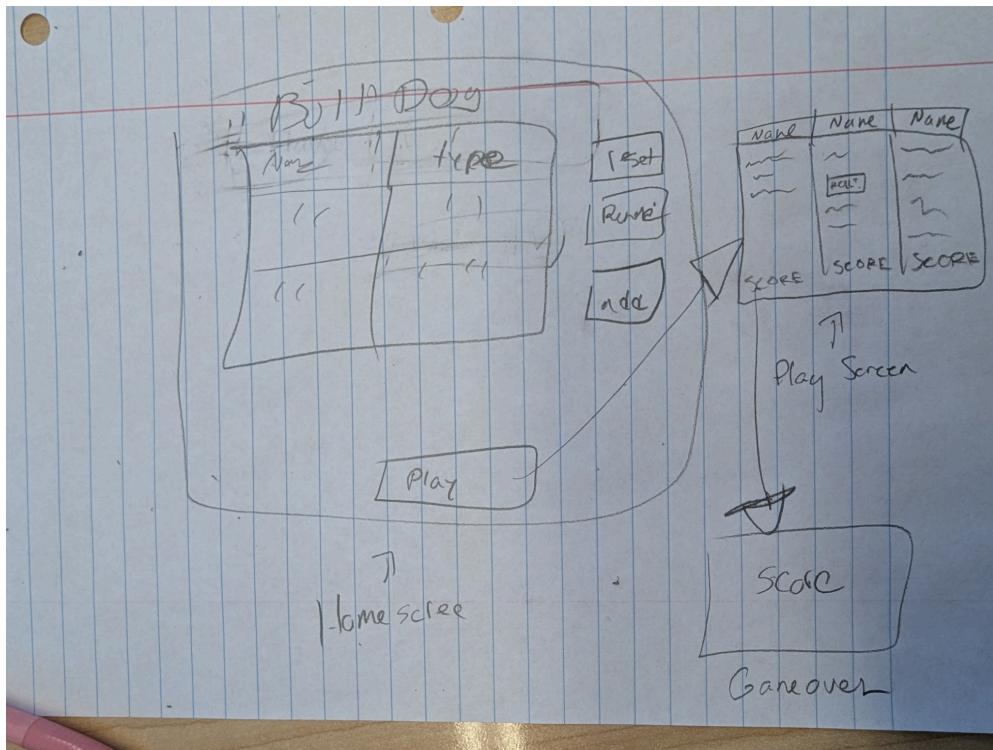
COS 420

Program 4

2.21.2025

Lab Book

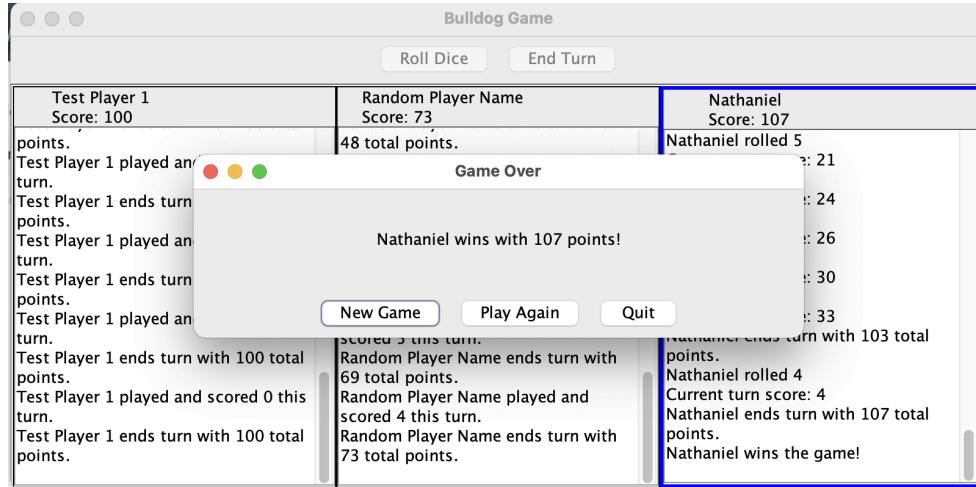
Time spent working with AI: February 20, 9:20 pm - 11:30 pm



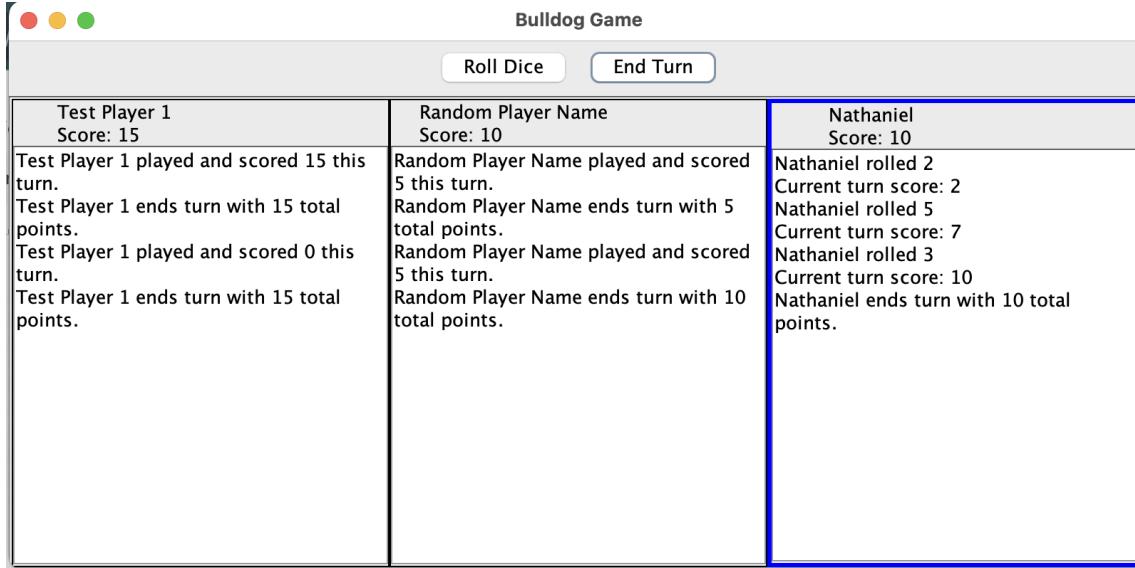
Concept Illustration was created by James Tedder, Courtney Jackson, and Nathaniel Serrano

For this project, I utilized the same AI tool as before, ChatGPT 4o. Working with ChatGPT 4o for this assignment gave me a lot of insights into what to expect when working with AI tools. Most notably, you will find that AI tools excel when they are not restricted by the user's own design philosophy but can struggle when the user wants something super specific. I began my prompting with a simple request: "Implement a GUI for this game (bulldog) utilizing JSwing." I intentionally chose to not get specific about design and layout until the end, after I had a base product working. What the tool produced was almost a fully functional program right off the bat. I had to poke it a bit and ask, "would any of the previous player classes require any changes?" so that it would offer to modify the HumanPlayer class that needed to interact with the GUI, which I then accepted. Next, after doing some testing I realized that the non-human classes still had to be rolled manually by the user hitting the roll button, and informed the AI tool of this. The tool then automated all players outside of HumanPlayer, as requested. The basics were complete. Now, I had to try and replicate the concept art from class. I chose to work in reverse, starting on the game over screen, which would show who won and their score. I prompted, "when there is a winner, create a popup window that shows the winner and their score, along with buttons that allow the user to quit the program or start a new game, which takes the user back to the creating players menu." Upon successful implementation of this, I realized that I could still press the 'roll' and 'end turn' buttons on the main screen, which added to my score and continued the game. To rectify this, I simply prompted, "can you disable the roll and end turn buttons after the game ends and the new window pops up?" which successfully implemented the feature. The game over screen was (mostly) complete. Later on, when adding some finishing touches, I implemented a play again button, though that was without the use of the AI tool. The

final screen looks a bit more complex than the original illustration, but the idea of it being a popup window is still very much present.

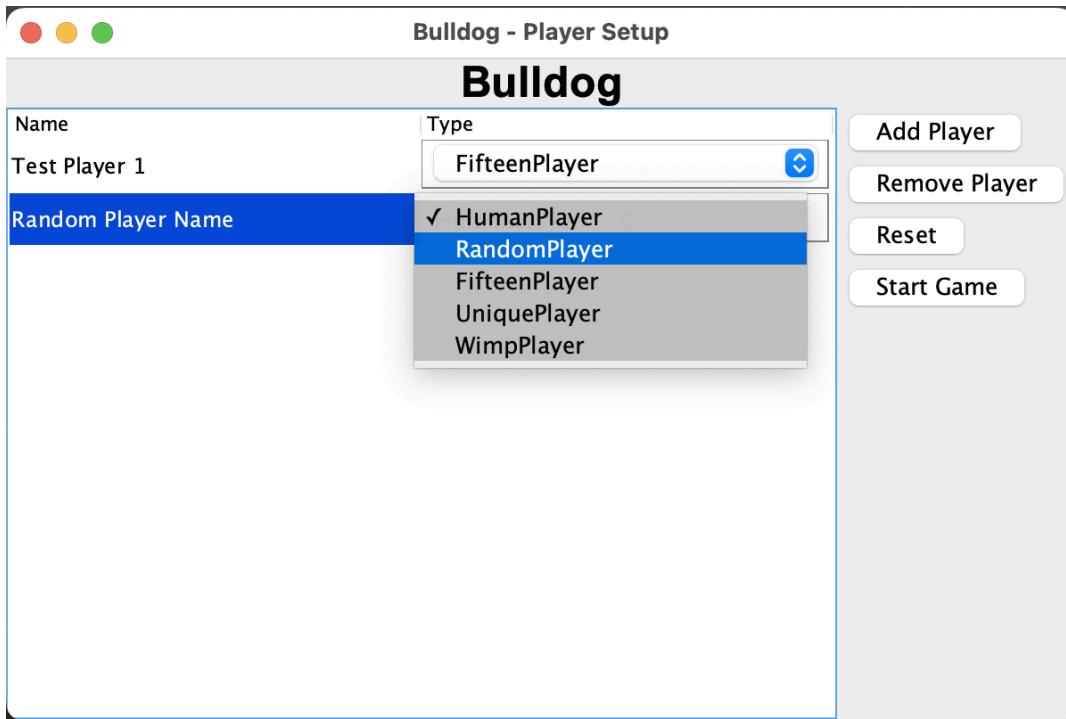


Next, I decided to work on the main game screen, where the AI tool originally had displayed a simple text log of every player's scores. This, however, did not align with our concept art for the game screen, which is notable for its column-like layout, where each player's scoring information is localized entirely in its own column. To achieve this using ChatGPT 4o, I prompted, “Next, rearrange the playscreen into a column layout where each player’s score and turn information will have a dedicated column. This screen will have at most 5 columns before implementing a side scroll function to view more columns if the number of players exceeds 5.” This prompt, while very specific, led to a good starting point for the game screen. After some troubleshooting and dealing with an issue caused by one of the tool’s methods, eventually a proper game screen was implemented, reminiscent of what was shown in the original concept art but with what I believe to be a slightly better design, an example being the static buttons at the top of the screen for all human players instead of them being in each column as depicted in the concept art.



Lastly, I worked on the main menu for the game, where the user can create players before starting the game. Originally, the AI tool had this work with a text box asking the user to enter the number of players, then each player's name and type, utilizing editable text boxes and buttons. To implement what is shown in the concept art, I had to ensure I was super specific in my prompt: "Instead of multiple popup windows for creating players, have one window where there are buttons to add a new player, remove a player, and reset (remove all players). On the bottom of the window should be a start game button. On the top of the window should be text that says the name of the game: Bulldog. The middle and middle-left sections of the window should show the list of players, where the name has to be entered, with each player having a dropdown menu to select the type of player." Again, a very specific prompt. In order to refine everything, I elaborated, "Upon selecting the add player button, a pop up window should appear asking for the name of the new player, which then appears on the screen." Lastly, I requested for the list of players to be formatted into a table, with one column being 'name', the other, 'type.' The latter would have a dropdown menu where the user can select the player type. Lastly, I had

the AI tool add some padding between the rows so the drop down menu was not cut off and added an error that would appear if the user tried to start a game with less than two players.



After the GUI was finally complete, all that was left was to implement documentation in the style of Javadoc. This required a basic prompt: “Lastly, take this java file and add documentation in Javadoc style.”

Essay

I mentioned it briefly before, but this assignment gave me a lot of insights in the strengths and weaknesses of AI tools. The tools are really strong if you do not care about the specifics in how things are done, merely the results. When it comes to something like designing a GUI, there are many subjective elements to it that can be difficult to relay to an AI tool. It was really easy to have ChatGPT 4o produce a working Bulldog program with a Java Swing GUI. Getting that done took around 15 minutes. Getting a Bulldog program that looked like the concept art we designed in class, however, is what took the extra two hours. It is important to note that my Java Swing knowledge is incredibly limited, mostly consisting of what was discussed in class. I have some web-design and front-end experience from past projects that tells me what things should look like, but when it comes to coding in Java Swing I am a total amateur. That being said, ChatGPT 4o did a perfectly fine job in producing GUIs that met the criteria I established. One thing to note is that there were numerous times when the tool would omit some methods from the class, requiring me to prompt it to add it back in. As long as I was paying attention it was not an issue. Overall, I am very impressed with the outcome of this assignment. I went in only really knowing how to add a button to a panel, but through the use of AI tools, was able to produce a fully functioning GUI with multiple screens. On top of that, spending time working with all that Java Swing code improved my own understanding of how it works. Of course, if I were an expert in Java Swing, I likely would have been able to get even more specific in my prompting, which would have sped up the time it took to complete this assignment by a large margin.